

UNITED STATES PATENT APPLICATION FOR

SYSTEM FOR PROVIDING FIREWALL TO A COMMUNICATION
DEVICE AND METHOD AND DEVICE OF SAME

Inventors:

PETER M. RIGSTAD
JOHN Y. YOON
MARK FARAHAR
BARANI SUBBIAH
BRANT D. THOMSEN
DAVID E. RICHMOND
VERED MAOR SEMEL
EARL BRYNER

Prepared by:

WAGNER, MURABITO & HAO LLP
TWO NORTH MARKET STREET
THIRD FLOOR
SAN JOSE, CALIFORNIA 95113
(408) 938-9060

3COM-3828.MCD.US.P JPW/RMP

SYSTEM FOR PROVIDING FIREWALL TO A COMMUNICATION DEVICE AND METHOD AND DEVICE OF SAME

RELATED APPLICATION

5 The following U.S. Patent is herein incorporated by reference as background material: United States Patent No. 5,968,176, issued October 19, 1999, entitled "MULTILAYER FIREWALL SYSTEM" to Nessett et al.

TECHNICAL FIELD

10 The present invention generally pertains to the field of data networking. More particularly, the present invention is related to a system for providing a hardware firewall for a device without such a firewall in a network where it is desirable that devices have such a firewall.

15 BACKGROUND ART

 When providing security for a network, one traditional method is a firewall at the perimeter at the network. However, it is desirable to allow authorized users to connect to the network remotely. For example, a corporation may wish to allow its employees to connect to a corporate
20 network from home. While a perimeter firewall provides protection to the network from unauthorized access from remote devices, it may not be effective to protect against a security breach originating from an authorized device. For example, an employee may present a security risk due to his home computer being compromised.

25

 One conventional method of providing security for a network is via software implemented firewalls. While software firewalls may be implemented on the devices that are physically remote from the network, the software firewalls are susceptible to attacks from Trojan programs and

other hacking methods. For example, the data may flow from a communication device providing the network interface to a host device's operating system software stack where the software firewall performs its rule checks to determine whether the data should proceed further up the software stack. (And for outbound data the software firewall again resides at a point well above the network interface.) Numerous examples have been reported in which such software firewalls have been compromised.

Thus, while a corporation may desire that its employees are able to access portions of the corporate network from home or elsewhere outside the office, this presents significant security concerns. Even if the corporation provides its employees with a software firewall for their home computers, an employee's computer may be compromised without the employee's knowledge by a Trojan program, for example. Furthermore, when the employee logs into the corporate network, the perimeter firewall inside the corporate network provides little security.

Other conventional methods provide for a hardware implemented firewall by implementing a firewall on a network interface card (NIC). The corporation may then provide each employee with such a NIC. So long as the employees use these NICs, the network may be protected better than with software firewalls. However, many individuals already have legacy NICs without such firewalls. If the employee uses such a legacy NIC to connect to the corporate network, corporate network security may be compromised as the employee's computer is left unprotected.

Thus, a need has arisen for a way to prevent unauthorized access to a network. A still further need exists for a method that provides protection for a network that has devices making remote or local connections. An even

3COM-3828.MCD.US.P JPW/RMP

SUMMARY

Embodiments of the present invention provide a way to prevent unauthorized access to a network. Embodiments provide protection for a network that has devices making remote and local connections.

- 5 Embodiments provide protection that is not easily defeated by hacking techniques such as Trojan programs.

10 A method, system, and device for providing security in a computing network are disclosed. One embodiment provides for a system having a server for distributing policies to be implemented by firewall devices in the network. The firewall devices provide hardware implemented firewalls to communication devices making network connections. The system has logic to allow a connection to be made to the network via a communication device at a node provided the firewall device is at that node. Therefore, the firewall
15 device must be in the system for a connection to be established via the communication device. Additionally, the system is configured to cause data transferred by the communication device to be processed by the firewall.

20 These and other advantages of the present invention will no doubt become obvious to those of ordinary skill in the art after having read the following detailed description of the preferred embodiments which are illustrated in the various drawing figures.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a diagram of a system with a device having an embedded firewall coupled to a host device, according to embodiments of the present invention.

FIGURE 2 is a diagram showing further details of a system with a device having an embedded firewall coupled to a host device, according to embodiments of the present invention.

FIGURE 3A and FIGURE 3B are diagrams illustrating a resource allocation before and after a swap, according to embodiments of the present invention.

FIGURE 4 is a diagram illustrating a device with an embedded firewall coupled to a device without such a firewall, according to embodiments of the present invention.

FIGURE 5 is a diagram illustrating a device with an embedded firewall coupled to a device without one, according to embodiments of the present invention.

FIGURE 6 is a diagram of a network stack with a driver for routing data to an embedded firewall to provide the same for a device without one, according to embodiments of the present invention.

FIGURE 7 is a diagram of a shim above a driver for routing data to an embedded firewall to provide the same for a device without one, according to embodiments of the present invention.

FIGURE 8 is a diagram of a shim below a driver for routing data to an embedded firewall to provide the same for a device without one, according to embodiments of the present invention.

5

FIGURE 9 is a flowchart illustrating steps of a process of configuring a firewall device for operation, according to embodiments of the present invention.

- 10 FIGURE 10 is a flowchart illustrating steps of a process of providing network security by adding an embedded firewall, according to embodiments of the present invention.

BEST MODE FOR CARRYING OUT THE INVENTION

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. While the invention will be described in conjunction with the preferred embodiments, it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications, and equivalents, which may be included within the spirit and scope of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be obvious to one of ordinary skill in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the present invention.

Some portions of the detailed descriptions which follow are presented in terms of procedures, logic blocks, processing, and other symbolic representations of operations on data bits within a computer memory.

These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. In the present application, a procedure, logic block, process, etc., is conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proved convenient at

times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as "measuring", "calculating", "receiving", "computing" or the like, refer to the actions and processes of a computer system, or similar electronic computing device. The computer system or similar electronic computing device manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission, or display devices. The present invention is also well suited to the use of other computer systems such as, for example, optical and mechanical computers.

Embodiments provide for a system that may be centrally managed and may have nodes with devices having hardware implemented firewalls. Referring now to Figure 1, a node 150 has a first device 120 (e.g., a firewall device 120) having a hardware implemented firewall 125. The first device 120 is coupled to a host device 130 (e.g., personal computer, laptop, personal digital assistant, etc.). The firewall device 120 may be implemented on a device such as a PCMCIA card, although the present invention is not limited to such a card. The host device 130 may be coupled to a second device 140, such as a network interface card (NIC), which provides a physical communication interface to a network 210. The second device 140 may be a

communications device without a hardware firewall. Throughout this application, second device 140 may be referred to as a communication interface device or communication device 140. The communication interface device 140 may connect to a server 160 via Ethernet. However, the present invention is not limited to Ethernet. As the system 170 does not require the server 160, the node 150 may also be referred to as a system 170.

Still referring to Figure 1, the firewall device 120 has logic 135 to allow the node 150 to establish a connection to the network 210 via the communication interface device 140. For example, the firewall device 120 may implement hardware token authentication. Alternatively, the firewall device 120 may connect to another device, such as a token 117. The firewall device 120 also may have a configuration integrity checker 145a for checking integrity of software components in said system. A portion of the configuration integrity checker 145b may reside on the host device 130.

The system 170 also has a server 160 that may store policies to be transferred to nodes 150 and implemented by a firewall device 120 at a node 150. This server 160 may be referred to as a policy server 160. It will be understood that the policy server 160 is not required; for example, the firewall device 120 may store policies.

Furthermore, the node 150 is configured to cause data transferred by the communication interface device 140 to be processed by the firewall 125. For example, any data that is received by the communication interface device 140 is processed by the firewall 125 and any data that is to be sent to the network 210 via the communication interface device 140 is also processed by the firewall 125. In either case, the firewall 125 processing may occur either before or after the communication interface device 140 has

the data. Embodiments described herein provide for suitable techniques for having all data transferred by the communication interface device 140 to be processed by the firewall 125. However, the present invention is not limited to the described embodiments.

5

Embodiments provide additional features to the system 170, as shown in Figure 2. The firewall device 120 may store one or more addresses 235 of policy servers 160, which the firewall device 120 may try to find when it comes up. The policy servers 160 may be administered by an administrator console (not shown) that defines the firewall rules. Thus, the firewall device 120 may store policies 275 consisting of various rules defining: protocols it accepts or rejects, types of IP (Internet Protocol) addresses to which it is allowed to talk, etc. The administrator console may define these rules and provide them to a policy server 160, which gives them securely to the firewall device 120.

The firewall device 120 may receive updates to the policies 275 from the policy server 160. If the firewall device 120 cannot find a policy server 160, then the firewall device 120 may rely on fallback policies 275 that are stored on the firewall device 120 and/or another device, such as, for example a token 117. Multiple fallback policies 275 may be stored for one or more users. The firewall device 120 has stored therein rules which are used to determine which policies 275 to use depending on the type of communication the communication interface device is using and/or location. In one embodiment, the policy servers 160 are not used. Instead the host device 130 may be used as an administrator.

The transmissions between the firewall device 120 and the policy server 160 may be encrypted to provide additional security. The firewall

device 120 may store a key, certificate, or the like 245, which is used to encrypt/decrypt the data that is transferred. Thus, the firewall device 120 is also shown with policy server communication logic 250, an encryption engine 252, and a cryptographic hash engine 254. The data that passes
5 through the host device 130 network stack 265 to or from the communication interface device 140 may be encrypted and may not be decrypted by anything other than the firewall device 120. Throughout this application the term transfer security logic may be used to describe the components to provide additional security to the system by encrypting network transfers.

10 In order to provide additional security, embodiments provide for various logic to perform configuration integrity checking, which may be used to check if various software components in the system (e.g., in the host device 130) have been compromised. For example, embodiments may check
15 the integrity of software drivers (e.g., firewall device driver 280 or drivers in the network stack 265) that are used to route data to the hardware firewall 125. In one embodiment, the check is performed on all registered software components. A portion of this logic may reside on the host device 130. For example, the host device 130 is shown having a configuration integrity
20 checker validation plugin 286 and a configuration integrity checker engine 145b. Portions of this logic may reside on the firewall device 120. The firewall device 120 may have a configuration integrity checker (CIC) 145a comprising CIC engine validation logic 246, CIC component validation logic 247, and hardware driver validation logic 248. The CIC 145a may
25 examine memory of CIC engine 145b and low level drivers (e.g., 280) and perform a cryptographic hash of those drivers by reading the memory contents directly out of the host device 130 O/S memory space and onto the firewall device 120 and then compare them against a stored cryptographic hash value on the firewall device 120. The stored cryptographic hash value

may be distributed by a policy server 160 and potentially stored on the firewall device 120.

Still referring to Figure 2, the system also comprises logic 135 (e.g., authentication logic) that allows a connection to be made to the network 210 provided the firewall device 120 is in the system. Without this logic 135, an attempt to connect to the network 210 will be refused. An authentication server 260 may be used to configure and enforce authentication. In this fashion, the communication interface device 140 is prevented from establishing a connection to the network 210 unless the firewall device 120 is coupled to the host device 130. For example, the firewall device 120 may implement hardware token 295 authentication. Alternatively, the firewall device 120 may connect to another device which is a token 117. Authentication logic 135 may reside entirely on the firewall device 120 or a portion of it may reside on the firewall device 120 with the rest on a separate device. The host device 130 may contain a portion of the authentication logic 135h. The firewall device 120 may store therein keys, policies 275, data, etc., that are used in configuring communication connections (e.g., a connection via communication interface device 140). In this fashion, a connection may not be made by the communication interface device 140, unless the firewall device 120 is present and operational. Thus, if a user removes the firewall device 120, the user may not use the communication interface device 140 to connect to the network 210. However, the user may still be able to use the communication interface device 140 to connect to other networks that do not require the firewall device 120 to be in the system in order to establish a connection. Thus, for example, a corporation may be able to enforce a requirement that employees use the hardware firewall 125 when connecting to the corporate network 210. An authentication server 260 may be contacted in this process. In one embodiment, the Extensible

Authentication Protocol is used to authenticate PPP connections between the host device 130 and a RADIUS server. This may be used for a variety of connections including, e.g., Ethernet, WLAN, modem, and Virtual Private Networks (VPN).

5

Additionally, the authentication may be tied to the CIC 145a. For example, the firewall device 120 may first perform a configuration integrity check. The firewall device 120 only passes the information needed for authentication (e.g., certificate) if the CIC check determines that the integrity is good.

10

The system 170 may also require the firewall device's 120 presence for O/S login. If someone pulls out the firewall device 120, they are automatically logged out. For example, if the CIC 145a or 145b determines that the firewall device 120 is pulled out, they are automatically logged out or cannot log in.

15

The system 170 may also comprise an alert log 297 for logging security alerts, which may be detected by the CIC 145a or by the firewall 125. The policies 275 may describe which events are to be logged. When such an event happens, an alert is created. If the host device 130 is connected to a policy server 160, then the alert may be sent to the policy server 160. Alerts may also be sent to other servers. Optionally, the alert may be stored even if it is transferred to a server 160. If no connection exists to an alerting system, then the alert is preferably stored. Then, the next time the firewall device 120 has access to an alerting service it may transfer the alert log to that server 160. In one embodiment, the data is sent LIFO so that the most recent alerts are received first. The policies 275 may also contain information that indicates whether an alert should be notified on the client

20

25

130. While a remote alert service is used in some embodiment, a remote alert service is not required.

The system 170 may also display the alerts to the host device user.

5 Thus, one embodiment provides for a graphical user interface (GUI not shown), which is driven by the GUI interface layer 298.

In order to process network data with the hardware firewall 125, embodiments provide for various techniques with which to transfer or route
10 the data to the firewall device 120. For example, the system is configured to cause data transferred by the communication interface device 140 to be processed by the firewall device 120. Some of the techniques are suitable for a wide variety of connection types (e.g., Ethernet, WLAN, VPN, modem, etc.). Others may be limited in the types of connections they support.

15

Referring now to Figure 3A and Figure 3B, one embodiment swaps host device 130 O/S resource spaces between the communication interface device 140 and the firewall device 120. Thus, in Figure 3A, resources A 340
20 are originally assigned to the communication interface device 140 and resources B 320 are originally assigned to the firewall device 120. The dashed lines between the host device 130 and the devices 120, 140 indicate how the resources are allocated. The solid lines indicate connections 350 for actual data transfers. After swapping as shown in Figure 3B, resources A 340 are now assigned to the firewall device 120 and resources B 320 are now
25 assigned to the communication interface device 140.

In the present embodiment, the flow of data may be from the network 210 to the communication interface device 140 to the firewall device 120 to be processed with the hardware firewall 125. Then, the processed data may be

transferred from the firewall device 120 to the host device 130. Because the resources of the communication interface device 140 and the firewall device 120 have been swapped, the host device 130 O/S believes the data came from the communication interface device 140. The swapping of the resources
5 may be implemented via software.

Figure 3A and Figure 3B show a data transfer connection 350 between the firewall device 120 and the communication interface device 140 for transferring data between the devices 120, 140. This may be a physical
10 link (e.g., PCMCIA, etc.), wireless, infra red, etc. Also shown are data transfer connections for transferring data between the devices 120, 140 and the host device 130. It will be understood that not all of the data transfer connections 350 shown may be needed to effect the necessary data transfers. The data may be transferred between the communication interface device
15 140 and the firewall device 120 in any suitable fashion. Because there may not be a standard for transferring data between the communication interface device 140 and the firewall device 120, a non-standard solution may be used.

20 Still referring to Figure 3A and Figure 3B, a reverse scenario is also possible for outbound data. When the client device 130 O/S has data to go onto the network 210 via the communication interface device 140, it transfers it to what it believes is the resource space of the communication interface device 140. However, because the resources spaces have been
25 swapped, this is now the resource space for the firewall device 120. Thus, the firewall device 120 receives the data, processes it with the hardware firewall 125 and transfers it to the communication interface device 140.

Referring now to Figure 4, another embodiment for providing network data to the firewall device 120 is shown. Thus, another embodiment for causing data transferred by the communication interface device 140 to be processed by the firewall device 120 is shown. In this embodiment, a physical connection 410 is made between the communication interface device 140 and the firewall device 120. The firewall device 120 also has a physical connection to the network 210. The physical connection 410 between the two devices 120, 140 may be the same medium as the network connection. For example, if the communication interface device 140 is connecting to a LAN via an Ethernet cable, then such a cable may be used. However, the present embodiment is not limited to using an Ethernet cable.

It will be understood that the firewall device 120 may be coupled between the communication interface device 140 and the host device 130, as well. The location of the firewall device 120 may be selected to provide the protection desired. Thus, in this embodiment, all data that is processed by the communication interface device 140 is also available to the firewall device 120 for processing. Furthermore, received data may be processed by the firewall 125 before it enters the host device 130 and sent data may be processed by the firewall 125 after it leaves the host device 130.

Another embodiment for providing network data to the firewall device 120 (e.g., causing data transferred by the communication interface device 140 to be processed by the firewall device 120) is shown in Figure 5. In this embodiment, the firewall device 120 and the communication interface device 140 are coupled together by, for example, an MPCPI adapter (Mini Peripheral Component Interconnect). Thus, the firewall device 120 may be plugged into the top of the communication interface device 140.

Alternatively, the firewall device 120 may be slid into the top of the communication interface device 140. As shown, the firewall device 120 is physically connected to the network 210. However, the communication interface device 140 could be physically connected to the network 210 instead, with the firewall device 120 receiving the network data from the communication interface device 140.

Another embodiment for providing network data to the firewall device 120 (e.g., causing data transferred by the communication interface device 140 to be processed by the firewall device 120) is shown in Figure 6. In this embodiment, a driver 610 for the communication interface device 140 has properties which allow it to transfer or route the data to the firewall device 120. The present embodiment may be suitable for a wide variety of connection types. The communication interface device driver 610, which may be at the physical layer 615, is aware of the firewall device 120. Thus, data received from the network 210 goes from the communication interface device 140 to the communication interface device driver 610 to the firewall device 120. Arrows between the devices 120, 140 and the communication interface device driver 610 show logical transfers. It will be understood that the data may pass through additional components, such as, for example, a firewall device driver 280. After the firewall device 120 uses the hardware firewall 125 to process the data, it may send it back to the communication interface device driver 610 for it to transfer up the data stack 265, a portion of which is shown in Figure 6. For example, the data may go through the data link layer 620 and the network layer 630.

Still referring to Figure 6, a reverse scenario is also possible. For example, data to be transferred out of the network 210 is first received by the communication interface device driver 610 and then transferred to the

firewall device 120. After receiving the data back from the firewall device 120, the communication interface device driver 610 passes it down to the communication interface device 140. In this fashion, all network data involving the communication interface device 140 is processed by the hardware firewall 125 in the firewall device 120.

Still referring to Figure 6, the communication interface device driver 610 may be designed to function with or without the firewall device 120. If the user attempts to connect to the network 210, embodiments require the presence of the firewall device 120 to access the network 210. Thus, the communication interface device driver 610 looks for the firewall device 120. If, however the user is connecting to a network that does not require the presence of the firewall device 120, then the communication interface device driver 610 does not look for the firewall device 120 and functions as a driver for only the communication interface device 140 would.

Referring now to Figure 7, yet another embodiment for providing network data to the firewall device 120 (e.g., causing data transferred by the communication interface device 140 to be processed by the firewall device 120) is shown. In this embodiment, a shim 710 is provided above the communication interface device driver 610. Thus, the original communication interface device driver 610 need not be replaced in this embodiment. The shim 710 may transfer data received from the communication interface device driver 610 to the firewall device 120 for firewall 125 processing. And the firewall device 120 may transfer processed data back to the shim 710 to be sent up the stack 265. The process may be reversed for data being sent out to the network 210. The arrow between the firewall device 120 and the shim 710 and the arrow between the communication interface device driver 610 and the communication device

140 illustrate logical transfers. In one embodiment, the shim 710 resides above a miniport driver in the data stack 265.

Referring now to Figure 8, yet another embodiment for providing
5 network data to the firewall device 120 (e.g., causing data transferred by the communication interface device 140 to be processed by the firewall device 120) is shown. In this embodiment, a shim 710 is provided below the communication interface device driver 610. Data may be transferred between the shim 710 and the firewall device 120 to allow firewall 125
10 processing of all network data for the connection used by the communication interface device 140. In this embodiment, the shim 710 talks directly to the hardware, therefore the shim 710 must know how to talk to the particular communication interface device 140 being used. The arrows between the firewall device 120 and the communication interface device driver 610 and the shim 710 illustrate logical transfers.
15

An embodiment provides for a method of configuring a firewall device 120 for operation in a network 210. Referring now to Process 900 of Figure 9, in step 910, a configuration integrity check of a software
20 component (e.g., firewall device driver 280, communication interface device driver 610, shim 710) is performed. For example, a cryptographic hash is performed on the software component to produce a hash value. The hash value may be compared to a hash value stored on the firewall device 120 to determine whether the software component has been compromised. Step
25 910 may be repeated at any time to assure that the configuration remains valid and that software components have not been tampered with.

Step 920 represents a branch depending on the result of the configuration integrity test. If the configuration integrity check fails, an

alert may be sent in step 925. For example, the firewall device 120 sends an alert to a policy server 160. However, the alert may be sent to any other server. Furthermore, the alert need not be sent. Alternatively, step 930 is taken instead, in which external communication is either shut down or prevented from being established by the host device 130.

In step 940, the alert may be stored on the firewall device 120. This may be the case whether the alert was sent to a server or not.

In step 950, an alert may be displayed to the user of the host device 130 via the GUI interface layer 298 causing an alert to be displayed on the host device 130. For example, a message may be displayed on a computer screen (not shown). Alternatively, a visual or audio warning signal may be triggered. For example, an LED may be lit.

If the configuration integrity test passes, then in step 960 a secure connection to the network 210 is established provided the firewall device 120 is coupled to the host device 130. For example, the host device 130 requests authentication information from the firewall device 120. If the firewall device 120 is not coupled to the host 130, the connection to the network 210 cannot be established as the needed connection authentication information is securely stored on firewall device 120.

In step 965, after a secure connection has been established, the firewall device 120 contacts the policy server 160 for policies 275. Alternatively, the firewall device 120 uses policies 275 that it has stored. For example, the policy server 160 may not be visible, in which case stored policies are relied on.

In step 970, the policy server 160 sends the policies 275 to the firewall device 120, which updates its stored policies 275. The firewall device 120 is now configured with the policies 275 to be used by the firewall 125 and the software components have checked out as being un-compromised.

5

In step 975, network data is checked against the policy rules and actions specified by the policies are performed. For example, data that is received by the communication device 140 is routed to the firewall device 120, according to any of the embodiments discussed herein. The Process 900 may then perform a configuration integrity check again.

10

Based on the outcome of checking the data against the policy rules and the configuration integrity check, steps 925-950 may be taken, in which security and/or configuration alerts are sent and/or stored and communication via the network 210 may be shut down. Process 900 may continue until communication is shut down or the network connection is otherwise terminated.

15

Process 1000 of Figure 10 illustrates one of the embodiments to provide a hardware implemented firewall 135 to a communication device 140 without such a firewall 135. Process 1000 may be a subset of Process 900 of Figure 9. For example, Process 1000 may be substituted for steps 960-975 of Process 900. In step 1010, a connection to a network 210 is allowed to be established when using a communication interface device 140 only if a firewall device 120 comprising a hardware implemented firewall 125 is coupled to a host device 130. For example, the host device 130 requests connection configuration information from the firewall device 120. If the firewall device 120 is not coupled to the host 130, the connection to the network 210 cannot be established as the needed connection authentication

20

25

information is securely stored on firewall device 120. The firewall device 120 may condition this transfer on the passing of a configuration integrity check, as in process 900 of Figure 9. The policies 275 that the firewall device 120 has stored thereon may also be used to determine whether the configuration information will be transferred to the host device 130.

In optional step 1020, resource spaces (320, 340) that are reserved for the communication interface device 140 and the firewall device 120 are swapped in the host device 130. Therefore, the host device 130 treats the communication interface device 140 as the firewall device 120 and vice versa.

In step 1030 data is received from the network 210 over the connection established via the communication interface device 140.

In step 1040, the data is routed or transferred to the firewall device 120 to be processed by the hardware implemented firewall 125. The routing may take place at a physical layer 615 of the host device stack 265 (e.g., by a communication interface device driver 610). However, the present invention is not limited to this method of transferring data to the firewall device 120. In other embodiments, the data is transferred to the firewall device 120 by a direct connection to the communication device 140 or by routing from a shim 710 in the data stack 265. If step 1030 is taken, step 1040 may comprise a transfer from the communication interface device 140 to the firewall device 120.

In step 1050, the firewall device 120 processes the data with the hardware implemented firewall 125.

In step 1060, the data is transferred from the firewall device 120 to the host device 130. The host device 130 may then transfer the data up the data stack 265. Process 1000 then ends. The data may be transferred from the firewall device 120 to the host device 130 by various techniques described herein. For example, the techniques described in conjunction with Figure 3A - Figure 8 may be used. However, the present invention is not limited to these techniques.

It will be understood that Process 1000 of Figure 10 may be modified for data transfers going out to the network 210. For example, the data may be routed or transferred to the firewall device 120 before processing by the hardware implemented firewall 125, as discussed in conjunction with Figure 3A though Figure 8.

Therefore, it will be seen that embodiments of the present invention provide for a system, method, and device for preventing unauthorized access to a network. Embodiments provide protection for a network that has devices making remote connections. Embodiments provide protection that is not easily defeated by hacking techniques such as Trojan programs.

The foregoing descriptions of specific embodiments of the present invention have been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed, and obviously many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the

scope of the invention be defined by the Claims appended hereto and their equivalents.

2025-04-24 14:00:00